|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | | |  |
|  | | |
|  | | |
| 方圆ok1 | | | |
|  | **S Series Switches** | | 附件1-16K |
|  | |
| **Issue** | **V2.0** |
| **Date** | **2015-12-02** |
|  | |
| HUAWEI TECHNOLOGIES CO., LTD. | |
|  | | |

|  |
| --- |
| Copyright © Huawei Technologies Co., Ltd. 2015. All rights reserved.  No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.  Trademarks and Permissions  附件3-图 and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.  All other trademarks and trade names mentioned in this document are the property of their respective holders.  Notice  The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.  The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied. |

|  |  |
| --- | --- |
| Huawei Technologies Co., Ltd. | |
| Address: | Huawei Industrial Base  Bantian, Longgang  Shenzhen 518129  People's Republic of China |
| Website: | <http://e.huawei.com> |

iStack Technology White Paper

* + - 1. Keywords:

iStack, stack, topology collection, role election, high reliability, redundancy backup

* + - 1. Abstract:

Intelligent Stack (iStack) technology is a stacking technology that virtualizes multiple physical devices into one logical device for management and use. This document describes how iStack is implemented and how it is used on networks.

* + - 1. Acronyms and Abbreviations

|  |  |
| --- | --- |
| Abbreviation | Full Name |
| iStack | Intelligent Stack |

Contents

[1 iStack Overview 1](#_Toc436848480)

[1.1 Background 1](#_Toc436848481)

[1.2 Technical Advantages 2](#_Toc436848482)

[2 Implementation 4](#_Toc436848483)

[2.1 Concepts 4](#_Toc436848484)

[2.2 iStack Setup 5](#_Toc436848485)

[2.2.1 Physical Connection Setup 6](#_Toc436848486)

[2.2.2 Role Election 7](#_Toc436848487)

[2.2.3 Topology Collection 7](#_Toc436848488)

[2.2.4 Stable Running 7](#_Toc436848489)

[2.3 iStack Management 8](#_Toc436848490)

[2.4 iStack Maintenance 8](#_Toc436848491)

[2.4.1 Addition of New Member Switches 8](#_Toc436848492)

[2.4.2 Removal of Member Switches 9](#_Toc436848493)

[2.4.3 Topology Update 9](#_Toc436848494)

[2.5 Member Switch Intelligent Upgrade 10](#_Toc436848495)

[2.6 iStack Merge 10](#_Toc436848496)

[2.7 iStack Split and Multi-Active Detection 11](#_Toc436848497)

[2.7.1 iStack Split 11](#_Toc436848498)

[2.7.2 MAD 12](#_Toc436848499)

[2.7.3 MAD Multi-Active Handling and Fault Recovery 14](#_Toc436848500)

[3 iStack Packet Forwarding Principles 16](#_Toc436848501)

[4 Typical Networking Applications 20](#_Toc436848502)

[4.1 Simplified Network Management and Operation 20](#_Toc436848503)

[4.2 Increased System Access Capacity 21](#_Toc436848504)

[4.3 Long-Distance iStack 21](#_Toc436848505)

[4.4 Reduced Costs by Using Electrical Ports for Stacking 22](#_Toc436848506)

Figures

[**Figure 1-1** iStack networking 2](#_Toc436848399)

[**Figure 2-1** Stack virtualization of fixed switches 4](#_Toc436848400)

[**Figure 2-2** Stack setup process 5](#_Toc436848401)

[**Figure 2-3** Stack physical connections 6](#_Toc436848402)

[**Figure 2-4** iStack split 10](#_Toc436848403)

[**Figure 2-5** Original master and standby switches in the same iStack after a split 11](#_Toc436848404)

[**Figure 2-6** Original master and standby switches in different iStacks after a split 12](#_Toc436848405)

[**Figure 2-7** MAD through dedicated direct links between member switches and an intermediate device 13](#_Toc436848406)

[**Figure 2-8** MAD through full-meshed connections between member switches 13](#_Toc436848407)

[**Figure 2-9** MAD in relay mode 14](#_Toc436848408)

[**Figure 2-10** iStack split and multi-active handling 15](#_Toc436848409)

[**Figure 3-1** Unicast packet forwarding within a member switch 17](#_Toc436848410)

[**Figure 3-2** Unicast packet forwarding among member switches 18](#_Toc436848411)

[**Figure 3-3** Broadcast packet forwarding 19](#_Toc436848412)

[**Figure 4-1** Simplified network management and operation 20](#_Toc436848413)

[**Figure 4-2** Improved system access capacity by capacity expansion 21](#_Toc436848414)

[**Figure 4-3** Long-distance iStack networking 22](#_Toc436848415)

[**Figure 4-4** Using electrical ports for stacking 23](#_Toc436848416)

# iStack Overview

## Background

Currently, two models of Huawei switches are available on networks: fixed switches and modular switches.

* Fixed switches are often deployed on the network access layer or aggregation layer that does not require high reliability. A single fixed switch cannot provide flexible port and bandwidth capacity expansion, while adding new fixed switches will change the existing networking. The obvious advantage of fixed switches is their lower costs.
* Modular switches are often deployed on the network core or aggregation layer and have high reliability, performance, port density, and scalability. As these switches have high costs, it is not suitable to deploy them on the edge network close to the user side.

Intelligent Stack (iStack) combines the advantages of fixed and modular switches. iStack virtualizes multiple physical devices into one logical device (or an iStack) by connecting dedicated stack ports or service ports, as shown in Figure 1-1. You can manage this virtual device to manage all the devices in the iStack. The logical device has low costs of fixed switches and high scalability and reliability of modular switches. Huawei devices support two stacking modes:

* Stack card stacking

In this mode, dedicated stack ports on stack cards are connected to set up an iStack. This stacking mode does not require stack ports to be configured to set up an iStack.

* Service port stacking

In this mode, service ports are used to set up an iStack. This stacking mode does not require dedicated stack cards to be used to set up an iStack and support long-distance stacking.

iStack networking



## Technical Advantages

iStack has the following advantages:

* Simplified configuration and management

After an iStack is set up, multiple physical devices are virtualized into one logical device. You can log in to the iStack from any member device to uniformly configure and manage all the member devices. These member devices share the same IP address for login.

* Simplified network running

iStack virtualizes multiple devices into one logical device, simplifying the network. This network does not require ring network protocols xSTP, ERPS, RRPP, and SEP and gateway redundancy protocols such as VRRP to ensure network redundancy backup. iStack simplifies network configurations and supports inter-device link aggregation to achieve fast convergence and high reliability.

* High reliability

Redundancy backup is implemented between multiple member devices in an iStack. Additionally, an iStack supports inter-device link aggregation to implement inter-device link redundancy backup. Such device-level and link-level redundancy backup ensures uninterrupted service forwarding when some ports or member devices are faulty.

* High scalability

The network administrator can increase ports, bandwidth, and processing capacity of an iStack by simply adding member switches to the iStack.

* Flexible stacking mode

iStack supports stack card stacking and service port stacking. Stack card stacking is easy to deploy and operate without requiring member devices to be configured. An iStack can be set up when stack ports on stack cards are connected using dedicated stack cables. Service port stacking allows flexibly selecting the number of stack physical member ports (configured from service ports) according to actual service bandwidth to provide long-distance stacking. Service port stacking supports both optical port stacking and standard Ethernet RJ45 electrical port stacking. Electrical port stacking uses standard network cables for connection and so has lower costs.

* Lower costs

During initial network deployment, a small number of access users required only a few access devices to support. As services develop, more ports and higher bandwidth are required. iStack can support this requirement. It can easily expand the network access capability without changing the initial network plan and reduce initial investments.

# Implementation

## Concepts

Each device in an iStack is a member device and plays one of the following roles according to its functions:

* Master switch: manages the entire stack. An iStack has only one master switch.
* Standby switch: provides backup to the master switch. When the master switch fails, the standby switch will become the new master switch to take over all the services. An iStack has only one standby switch.
* Slave switch: forwards services. More slave switches ensure high stack forwarding capability. An iStack consists of one master switch, one standby switch, and slave switches.

Stack virtualization of fixed switches



In , the logical device virtualized by multiple fixed devices functions like a modular distributed device. The master switch in an iStack functions like the active MPU of the logical device, the standby switch functions like the standby MPU, and slave switches Slave1 and Slave2 function like LPUs. The master and standby switches can also function like LPUs to forward packets using service ports.

## iStack Setup

An iStack is set up after the following stages:

* Physical connection setup

When multiple switches are connected in a specific topology using an appropriate method based on network requirements, an iStack network is established.

* Role election:

Member switches send stack competition packets to one another to elect the master, standby, and slave switches according to stack election rules. The role election stage is in Electing state.

* Topology collection

The master switch collects topology information from all member switches and assigns stack IDs to the member switches. The topology collection stage is in Collecting state.

* Stable running

The master switch synchronizes the topology of the entire iStack to all the member switches, and member switches synchronize their system software and configuration files with the master switch. After that, the iStack runs stably. The stable running phase is in Running state.

Figure 2-2 shows the iStack setup process. After member switches are physically connected, the iStack setup process consists of three phases: role election, topology collection, and stable running, which are described in subsequent sections.

Stack setup process



### Physical Connection Setup

Member devices must be physically connected to set up an iStack. Stack ports are logical ports, while physical ports used for stack connections are stack physical member ports. In stack card stacking mode, ports on stack cards are used as dedicated stack physical member ports, removing the need to configure stack physical member ports. In service port stacking mode, service ports need to be configured as stack physical member ports and added to stack ports. A stack port may correspond to one stack physical member port or contain multiple stack physical member ports for link backup. One switch has only two stack ports, Stack-Port0 and Stack-Port1. Stack-Port0 and Stack-Port1 are sometimes called leaf port and right port for easy description.

Stack physical member ports are connected using dedicated stack cables, fibers, or standard network cables. Dedicated stack cables do not need to be configured, facilitating iStack setup. Physical devices that are far from one another can be connected using fibers as a virtual device. Member switches can be connected through Ethernet electrical ports using standard network cables to set up an iStack, allowing for a stacking distance of at most 100 meters, ensuring flexible iStack setup.

Stack physical connections



Stack physical member ports can be connected to set up an iStack in a chain or ring topology.

* Chain topology

The left/right interface of one device and right/left interface of another device are connected using stack cables, and all devices are connected in this way except that the first device is not connected to the last device. This connection mode is chain connection, as shown in Figure 2-3. The chain topology applies to long-distance stacking because the first and last devices do not need to be physically connected. The disadvantage of this topology is that the iStack splits if one link in the topology fails.

* Ring topology

If the right/left interface of the first device is connected to the left/right interface of the last device in the chain connection, a ring topology is formed, as shown in Figure 2-3. The ring topology is more reliable than the chain topology. When an iStack link in a ring topology fails, the iStack can still work properly and data can still be forwarded according to the shortest path, thus improving iStack link bandwidth efficiency.

### Role Election

An iStack consists of multiple member devices, each of which plays a role: master, standby, or slave switch. The process of determining member device roles is role election.

Role election occurs in case of topology changes, such as iStack setup, addition of new devices, iStack split, and iStack merge. The master switch is elected according to the following rules in the listed order (the election ends when a winning switch is found):

* The switch that is running for the longest time becomes the master switch.
* The switch with the highest stack priority becomes the master switch.
* The switch with the smallest bridge MAC address becomes the master switch.

After the master switch is elected, it collects topology information from all the member switches, calculates forwarding entries and blocking ports based on collected topology information, and sends the calculated information to all the member switches. The master switch also assigns a stack ID to each member switch. Then the standby switch is elected as the backup of the master switch. The switch that completes the startup first after the master switch is elected as the standby switch. If multiple switches complete the startup at the same time after the master switch, the standby switch is elected according to the following rules in the listed order (the election ends when a winning switch is found):

* The switch with the highest stack priority becomes the standby switch.
* If the switches have the same stack priority, the one with the smallest MAC address becomes the standby switch.

The member switches, except for the master and standby switches, function as slave switches to join the iStack.

### Topology Collection

Each member switch in the iStack exchanges Hello packets with its immediately-neighboring member switches to collect neighbor connection relationships. Hello packets carry topology information, including stack port connection relationships, stack ID, stack priority, MAC address, and running status of member switches.

After the master switch is elected, other member switches initiatively send topology information collected by themselves to the master switch. Topology information carries basic information and locations of these member switches. The master switch then collects topology information from all the member switches, calculates forwarding entries and blocking ports based on collected topology information, and sends the calculated information to all the member switches. The master switch also assigns a stack ID to each member switch.

### Stable Running

After role election and topology information collection are complete, all the member switches synchronize their system software and configuration files with the master switch.

An iStack supports automatic software loading. Member switches do not have to run the same software version, and they can set up an iStack as long as the software versions running on the member switches are compatible with one another. If software versions running on the standby switch and slave switches are different from that on the master switch, these switches download the system software from the master switch, restart with the new system software, and rejoin the iStack.

The iStack also supports configuration file synchronization. The standby switch and slave switches can download the configuration file of the master switch and apply this configuration file. This mechanism enables member switches to work like a single device and ensures that the standby switch takes over all the services of the master switch when the master switch fails, ensuring normal network running.

## iStack Management

* iStack login

iStack login includes local login and remote login. Local login indicates that you log in through the console interface of any member switch. Remote login indicates that you log in through the management interface or another Layer 3 interface of any member switch using Telnet or SSH. You can use the remote login mode if there are reachable routes between the switch and your operation terminal. When you log in to an iStack, you actually log in to the master switch, regardless of which member switch you logged in to. The master switch issues the configurations to all the other member switches. In this way, resources of member switches are managed consistently.

* File system access

Access to a file system refers to the operations performed on a storage device, including file display, and file/directory creation, deletion, and modification. Stack IDs are used to identify and manage member switches in an iStack. Each stack member switch has a unique stack ID. Stack IDs are also used in interface numbers to help configure and identify interfaces on member switches. An interface number starts with its device stack ID. On a standalone switch, its interfaces are numbered in the slot ID/subcard ID/port sequence number format. The slot ID is fixed as 0. After the switch joins an iStack, its interfaces are numbered in the stack ID/subcard ID/port sequence number format. For example, an interface on a standalone switch is numbered GigabitEthernet0/0/1. After the switch joins an iStack and is assigned stack ID 2, the interface number changes to GigabitEthernet2/0/1. The stack ID is also used in file system management. For example, the path slot2#flash:/cfg.zip indicates that there is a **cfg.zip** file in the root directory of the flash memory on the member switch with the stack ID 2.

## iStack Maintenance

During iStack maintenance, the master switch needs to monitor addition and removal of member switches, collects new topology information in real time, and maintains the existing topology.

### Addition of New Member Switches

A new member switch can join a running stack. During iStack maintenance, the master switch continues collecting topology information. When a new switch joins the iStack, the master switch is elected according to the following rules:

* If the new switch has not joined any iStack, the switch is elected as a slave switch without changing the master and standby roles in the iStack. For example, the new switch with the iStack configured is power off and connected to an iStack using stack cables. After the new switch is powered on, it becomes a slave switch in the iStack.
* If the new switch has already joined an iStack (for example, the switch has the iStack function configured and connected to another iStack using stack cables), the two iStacks merge into a new iStack. In this situation, the two iStacks compete with each other. Member switches of the iStack that fails the competition need to be restarted and join the iStack that succeeds in competition as slave switches.

The new member switch that joins an iStack functions like an LPU of a modular switch.

A member switch may by manually added to an iStack or join an iStack after recovering from a device fault or link fault.

### Removal of Member Switches

A member switch can leave an iStack. During iStack maintenance, determine whether a member switch leaves an iStack according to the following rules:

* Generally, immediately-neighboring member switches periodically exchange Hello packets. If a member switch does not receive Hello packets from its neighbor within multiple consecutive intervals, the member switch considers that its neighbor has left the iStack. Subsequently, the neighboring member switch is removed from the existing topology.
* If a member switch finds that its stack port goes Down, it immediately notifies the master switch of the port Down event. The master switch then immediately recalculates the current topology without waiting for the expiry of Hello packets.

If the master switch leaves the iStack, the standby switch takes over all the services of the master switch. If the standby switch leaves the iStack, the master switch selects a slave switch as the standby switch. This situation is just like that the iStack loses the standby MPU and physical resources such as interfaces on this MPU. If a slave switch leaves the iStack, the iStack loses physical resources of one LPU.

A standalone switch will run independently after leaving an iStack. Multiple connected member switches will form two independent iStacks after leaving an iStack, which is called iStack split.

A member switch may be manually removed from an iStack to change the existing topology or removed because of a device fault or link fault.

### Topology Update

Topology changes indicate that the ring topology changes into a chain topology or the chain topology changes into the ring topology. For example, when a link fails, the ring topology may change into the chain topology. Before a new member switch is added to an iStack, the existing ring topology needs to be changed into a chain topology.

In the preceding situations, the topology changes will not change the master and standby roles in the iStack and just automatically change the forwarding path if required. In case of such topology changes, member switches can still work properly.

## Member Switch Intelligent Upgrade

iStack has automatic software loading function. When an iStack is set up or new member switches join an iStack, the standby and slave switches or new member switches check whether their software versions are the same as or compatible with that of the master switch. If not, they download the system startup file from the master switch, restart with the new system startup file, and rejoin the iStack.

## iStack Merge

Two iStacks in the running state can merge into one iStack. As shown in Figure 2-4, after two iStacks merge, the master switches of the two iStacks compete to be the master switch of the new iStack. After the new master switch is elected, the member switches originally belonging to the same iStack as this new master switch retain their roles and configurations, and their services are unaffected. Switches in the other iStack restart and join the new iStack as slave switches. The master switch assigns new stack IDs to these switches. Then these switches synchronize their configuration files and system software with the master switch. During this time, services on these switches are interrupted. An iStack splits because an iStack link or member switch fails. After the iStack link or member switch recovers, the two iStacks merge into one again.

The iStack merging process is similar to the process when a new member switch joins an iStack. For details, see . When two iStacks merge, a new master switch is elected between the two original master switches. The master switch in the iStack that enters the running state first becomes the new master switch. If the two iStacks enter the running state at the same time, the master switch is elected according to the same rules used when an iStack is set up.

iStack split



## iStack Split and Multi-Active Detection

### iStack Split

If you remove some member switches from a running iStack without powering off the switches or if multiple stack cables fail, the iStack splits into multiple iStacks. After the iStack splits into multiple iStacks, multi-active detection (MAD) and conflict processing need to be performed to ensure stable service running.

Depending on the locations of the master and standby switches after a split, either of the following situations occurs:

* If the original master and standby switches are in the same iStack after the split, the master switch calculates the stack topology by deleting topology information related to the removed member switches, and then synchronizes updated topology information to the other member switches. When removed member switches detect that the timeout timer for stack protocol packets has expired, the switches restart and begin a new master election.

As shown in Figure 2-5, the original master switch (SwitchA) and standby switch (SwitchB) are in the same iStack after the split. SwitchA deletes topology information related to SwitchD and SwitchE and synchronizes topology information to SwitchB and SwitchC. After SwitchD and SwitchE restart, they set up a new iStack.

Original master and standby switches in the same iStack after a split



* If the original master and standby switches are in different iStacks after the split, the original master switch selects a new standby switch in its iStack, calculates stack topology information, and synchronizes updated topology information to the other member switches. The original standby switch becomes the new master switch in its iStack, and then calculates stack topology information, synchronizes stack topology information to other the member switches, and selects a new standby switch.

As shown in Figure 2-6, the original master switch (SwitchA) and standby switch (SwitchB) are in different iStacks after the split. SwitchA specifies SwitchD as the new standby switch, recalculates the stack topology, and synchronizes new topology information to SwitchD and SwitchE. In the other iStack, SwitchB becomes the master switch. It then recalculates the topology, synchronizes topology information to SwitchC, and specifies SwitchC as the new standby switch.

Original master and standby switches in different iStacks after a split



### MAD

All the member switches in an iStack use the same IP address and MAC address (stack MAC address). Therefore, after the iStack splits, multiple iStacks may use the same IP address and MAC address. To prevent this situation, a mechanism is required to check for IP address and MAC address collision after a split and then shut down all service ports on the device with a lower stack priority for normal service forwarding.

Multi-active detection (MAD) is a protocol that can detect iStack split and multiple-master situations and take recovery actions to minimize impact of an iStack split on services. When an iStack splits due to a link failure, MAD provides split detection, multi-active handling, and fault recovery mechanisms to minimize the impact of an iStack split on services. If multiple master switches exist after the MAD-enabled iStack split, these switches determine whether their master or standby roles according to the information in their received MAD packets. After the master and standby switches are elected, all the ports except the reserved ports (such as service stack ports and console port) on the standby switch are shut down.

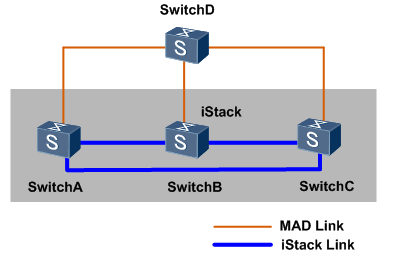
MAD can be implemented in direct or relay mode. The direct and relay modes cannot be configured together in the same iStack.

* MAD in direct mode

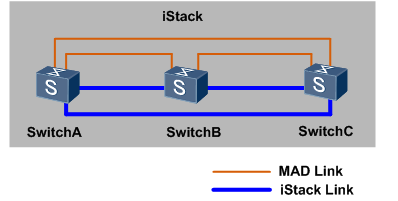
In direct mode, member switches use direct links over common network cables as dedicated MAD links. When the iStack is running normally, member switches do not send MAD packets. After the iStack splits, the member switches send a MAD packet every 1s over the MAD link to check whether more than one master switch exists.

In direct mode, member switches can be directly connected to an intermediate device or fully meshed to each other. As shown in Figure 2-7, each member switch has at least one MAD link connected to the intermediate device. In the full-mesh topology, member switches set up full-mesh connections through MAD links, as shown in Figure 2-8. That is, at least one MAD link is available between every two member switches.

MAD through dedicated direct links between member switches and an intermediate device



MAD through full-meshed connections between member switches



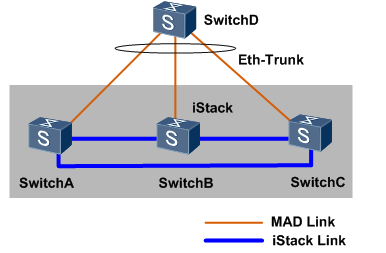
The use of an intermediate device can shorten the MAD links between member switches. This topology applies to iStacks where member switches are far from each other. The full-mesh topology prevents MAD failures caused by failures of an intermediate device, but full-mesh connections occupy many interfaces on the member switches. Therefore, this topology applies to iStacks with a few member switches.

* MAD in relay mode

In relay mode, MAD relay detection is configured on an inter-device Eth-Trunk interface in the iStack, and the MAD relay function is enabled on an agent. Every member switch must have a link to the agent and these links must be added to the same Eth-Trunk. Generally, an iStack connects to other devices using Eth-Trunk to ensure reliability. Compared with the direct mode, the relay mode does not require additional interfaces because the Eth-Trunk interface can perform MAD relay detection while running other services.

In relay mode, when the iStack is running properly, member switches send MAD packets at an interval of 30s over the MAD link and do not process received MAD packets. After the iStack splits, member switches send MAD packets at an interval of 1s over the MAD links to check whether more than one master switch exists.

MAD in relay mode



### MAD Multi-Active Handling and Fault Recovery

* MAD multi-active handling

After an iStack splits, the MAD mechanism sets the new iStacks to Detect or Recovery state. The iStack in Detect state can still work, and the iStack in Recovery state is disabled.

MAD handles a multi-active situation in the following way: When multiple iStacks in Detect state are detected by the MAD split detection mechanism, the iStacks compete to retain the Detect state. The iStacks fail in the competition enter the Recovery state, and all the physical ports except the reserved ports on the member switches in these iStacks are shut down. Then the iStacks in Recovery state no longer forward service packets.

As shown in Figure 2-10, an iStack splits into two iStacks, iStack1 and iStack2, because two iStack links are faulty. iStack2 fails in the competition with iStack1, changes to the Recovery state from the Detect state, and shuts down its uplink and downlink service ports. Subsequently, all services on iStack 2 are switched to iStack1 for forwarding.

iStack split and multi-active handling



* MAD fault recovery

After the faulty iStack links recover, the iStacks merge into one. The iStacks can merge in either of the following ways:

* When the fault iStack links recover, the iStacks in Recovery state restart and merge with the iStack in Detect state, and restore the shutdown service ports to Up state. Then the entire iStack recovers.
* If the iStack in Detect state is also faulty before the faulty iStack links recover, remove the iStack in Detect state first. Then use a command to start the iStacks in Recovery state, enabling the iStacks to take over services on the original iStack in Detect state. After the original iStack in Detect state and the faulty links recover, the iStacks can merge into one.

# iStack Packet Forwarding Principles

iStack uses distributed forwarding technology for Layer 2/Layer 3 packet forwarding, maximizing processing capacities of each member switch. Each member switch in an iStack has complete Layer 2/Layer 3 forwarding capabilities. During service forwarding, each member switch functions like an LPU of a modular switch and can learn the entire network’s MAC address table or FIB Layer 3 forwarding table. When a member switch receives Layer 2/Layer 3 packets that need to be forwarded, it looks up its local Layer 2/Layer 3 forwarding table to obtain the outbound interface of the packets and then sends the packets from this outbound interface. The outbound interface can belong to this switch or another member switch. Packet forwarding from the local member switch to another member switch is an internal forwarding behavior, just like a distributed modular switch forwards packets between cards through internal switch fabrics. That is, the number of hops for Layer 3 packets just increases by 1 regardless of how many member switches they pass through in an iStack, just like these packets pass through one network device.

When being forwarded within an iStack, packets carry the destination device’s stack ID in their headers and are forwarded from a proper outbound interface to the next member switch until they arrive at the destination member switch.

In an iStack set up using a chain topology, the forwarding path is fixed and packet forwarding is simple. In an iStack set up using a ring topology, inter-device packet forwarding is implemented based on the shortest path. When member switches join/leave the iStack or the link status changes, the system recalculates the topology; according to locations in the topology, other member switches calculate the forwarding path and blocking point with itself as the root and forward packets along the shortest path.

During inter-device unicast service forwarding in a ring topology, the source device’ stack port that is closest to the destination device is used as the outbound interface to forward packets, minimizing the number of intermediate devices between the source and destination devices. This mechanism ensures the shortest forwarding path. During broadcast packet forwarding (including unknown unicast and multicast packets) in a ring topology, a blocking point is set between the two member devices that are farthest from the source device to avoid broadcast packet loops and ensure shortest path forwarding.

As shown in Figure 3-1, the inbound and outbound interfaces of forwarded packets are located on the same member switch. When Slave1 receives packets, it loops up its forwarding table, finds that the outbound interface is a local interface, and sends the packets from this outbound interface.

Unicast packet forwarding within a member switch



As shown in Figure 3-2, the inbound and outbound interfaces of forwarded packets are located on different member switches. When Slave3 receives packets, it looks up its forwarding table, finding that the packets are destined for User2 connected to the standby switch. According to the shortest path algorithm, Slave3 uses its stack port 0 as the outbound interface of packets and forwards the packets to the master switch, as shown by the green arrows. The master switch then uses its stack port 0 to forward the packets to the standby switch. The standby switch looks up its routing table according to the destination address of the packets, finds that the destination host is a local user, and forwards the packets to User2 through its outbound interface.

Similarly, when Slave3 receives packets destined for User3 connected to Slave2, Slave3 uses its stack port 1 as the outbound interface and forwards the packets to Slave2, as shown by the purple arrows. Slave2 then forwards the packets to User3 through its outbound interface.

Unicast packet forwarding among member switches



Figure 3-3 describes broadcast packet forwarding (including unknown unicast and multicast packets) in an iStack. When Slave1 receives broadcast packets from the user side, it forwards the packets from two stack ports and sets a blocking point on the link that is farthest from Slave1. So, the blocking point of Slave1’s broadcast packets is located on the link between the master switch and Slave3. Similarly, when the master switch receives broadcast packets from the user side, it forwards the packets from two stack ports. The blocking point of the master switch’s broadcast packets is located on the link between Slave1 and Slave2 to prevent broadcast packets from being forwarded on the blocked link. This prevents loops.

Broadcast packet forwarding



# Typical Networking Applications

## Simplified Network Management and Operation

Using iStack to virtualize multiple devices on the same layer as one logical device can simplify the network structure and network protocol deployment as well as improve network reliability and manageability. Figure 4-1 shows common networking. This networking uses ring network protocols ERPS/MSTP and gateway redundancy protocols such as VRRP to ensure link backup and gateway backup. When protocol faults occur on the network, flappings may occur, complicating network deployment and maintenance. iStack and link aggregation can ensure network reliability without changing the existing network hierarchy, simplifying the network structure and reducing O&M costs.

Simplified network management and operation

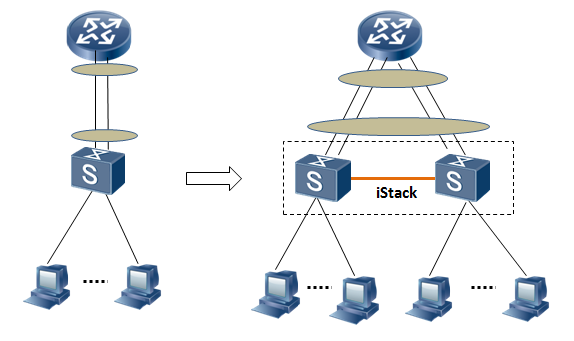


## Increased System Access Capacity

As services develop and access users increase, existing access devices cannot provide sufficient interfaces to support current service access requirements. To support service growth, change the single device as an iStack or add new devices to the existing iStack, as shown in Figure 4-2.

Adding member devices to an iStack increases access interfaces, and the iStack forwarding capacity doubles each time a member device is added to the iStack. Uplink interfaces are connected to upstream devices through inter-device link aggregation. When uplink interface bandwidth is insufficient, you can add member devices to increase the uplink interface bandwidth.

Improved system access capacity by capacity expansion



## Long-Distance iStack

iStack can use fibers to connect remote devices to set up a logical device. As shown in Figure 4-3, users at each floor connect to the external network through switches deployed at the corridors. These switches are connected using fibers to set up an iStack, as if there is only one access switch in each building. This implementation simplifies network structure. In addition, each floor is connected to the core network through multiple links, improving network robustness and reliability. The administrator only needs to configure iStacks and does not need to configure all the corridor switches one by one, reducing management and maintenance costs. The stacking distance varies only depending on optical modules and fibers. You can select optical modules and fibers according to network requirements to implement inter-floor, inter-building, and inter-area stacking.

Long-distance iStack networking



## Reduced Costs by Using Electrical Ports for Stacking

iStack service port stacking supports both optical ports and electrical ports. Electrical port stacking can be used for short-distance stacking within 100 meters. Such stacking mode uses only common network cables to connect electrical ports to set up an iStack, removing the need to use stack cards, optical modules, and fibers, as shown in Figure 4-4. This mode reduces costs and facilitates operations especially when many member switches and iStack links exist in an iStack.

Using electrical ports for stacking

